VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY
INSTITUTE OF MATHEMATICS

APLIKACE SHLUKOVÉ ANALÝZY NA REÁLNÝCH DATECH
APPLICATION OF CLUSTER ANALYSIS TO REAL DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Tomáš Onderlička**
AUTHOR

VEDOUCÍ PRÁCE — **RNDr. Libor Žák, Ph.D.**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav matematiky
Student:	**Tomáš Onderlička**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**RNDr. Libor Žák, Ph.D.**
Akademický rok:	2015/16

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Aplikace shlukové analýzy na reálných datech

Stručná charakteristika problematiky úkolu:

Bakalářská práce se bude zabývat využitím hierarchického a nehirarchického shlukování nad daty. Úkolem je najít vhodný počet shluků a jejich reprezentanty s ohledem na různé druhy podobnosti objektů a shluků.

Cíle bakalářské práce:

- popis základních shlukovacích metod
- nalezení možných shluků v reálných datech v různých programech
- využití nalezených shluků pro analýzu dat a další zpracování

Seznam literatury:

Anderberg, M. R. (1973): Cluster Analysis for Applications. Academic Press, New York

Lukasová, A.,Šarmanová, J. (1985): Metody shlukové analýzy. SNTL, Praha

Bezdek, J. C. (1981): Pattern Recognition with Fuzzy Objective Function Alghorithms. Plenium Press, New York

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2015/16

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

ABSTRAKT

Tato práce se zabývá hledáním podobných scénářů odpadového hospodářství získaných z optimalizačního nástroje NERUDA. K tomu jsou využity metody shlukové analýzy, pomocí kterých lze identifikovat podobné objekty a rozřadit je do skupin (shluků). Cílem práce je představit běžně používané algoritmy shlukové analýzy a vytvořit program, který tyto algoritmy implementuje. Vytvořeným nástrojem je poté provedeno shlukování reálných dat z nástroje NERUDA a následuje vyhodnocení kvality obdržených shluků.

KLÍČOVÁ SLOVA

shluková analýza, hierarchické shlukování, k-means, validace shluků, siluety

ABSTRACT

This bachelor's thesis deals with finding similar scenarios in the waste management acquired by an optimization tool NERUDA. Cluster analysis, a tool that identifies related objects and classifies them in groups (clusters), is used for this purpose. The aim of this thesis is to review basic algorithms of cluster analysis and to develop a software that implements them. The software is then used to cluster real data from NERUDA which is followed by an assessment of the obtained clusters.

KEYWORDS

cluster analysis, hierarchical clustering, k-means, cluster validation, silhouettes

ONDERLIČKA, Tomáš *Aplikace shlukové analýzy na reálných datech*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematického inženýrství, 2016. 35 s. Vedoucí práce RNDr. Libor Žák, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Aplikace shlukové analýzy na reálných datech“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Na tomto místě bych rád poděkoval RNDr. Liboru Žákovi, Ph.D. za odborné vedení této práce, Ing. Radovanu Šomplákovi, Ph.D. za cenné rady a mamince za příkladnou péči a podporu.

Brno ....

podpis autora

OBSAH

ÚVOD

Ústav procesního inženýrství (ÚPI) se dlouhodobě zabývá otázkou optimálního nakládání s odpadem a jeho energetickým využitím. Byl k tomu vyvinut optimalizační nástroj NERUDA, který minimalizuje celkové náklady spojené s rozvozem a zpracováním odpadu. Generuje tisíce scénářů, které je dále nutné analyzovat nástrojem PIGEON. Ten je ovšem výpočetně velmi náročný, proto není možné takto analyzovat každý scénář. Je tedy nutné jich vybrat pouze určitý počet.

Scénáře se od sebe často příliš neliší, proto se nabízí ty podobné identifikovat, spojit (shluknout) do skupin a poté vybrat vhodné reprezentanty těchto skupin. A právě tím se v této práci budeme zabývat. Používat k tomu budeme metody shlukové analýzy.

V následujícím textu si nejprve stručně představíme nástroje NERUDA a PIGEON. Dále se budeme zabývat metodami a algoritmy shlukové analýzy, které poté budeme aplikovat na výsledky nástroje NERUDA. V rámci této práce také vznikne software, pomocí něhož bude možné scénáře shlukovat, analyzovat obdržené výsledky a vybírat vhodné reprezentanty.

1 NÁSTROJE NERUDA A PIGEON

Základní myšlenka modelu NERUDA spočívá v tom, že se vlastník odpadu (obec) rozhoduje, jak s odpadem co nejlevněji naložit. O výši nákladů rozhoduje cena za zpracování v daném zařízení a cena dopravy. Z matematického hlediska se řeší tzv. logistická úloha.

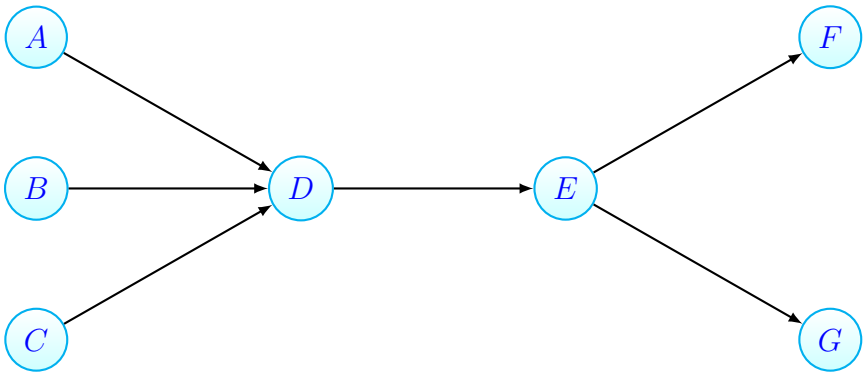
Nástroj se také zabývá:

- výběrem vhodné lokality pro výstavbu zařízení a jeho optimální kapacity,
- popisem toku odpadu v rámci sledovaného území,
- podporou návrhu logistického řetězce (svozová vozidla, překládací stanice, silniční a železniční doprava),
- hodnocení atraktivity záměrů - dostupnost odpadu a očekávaná cena za zpracování.

Kvůli neurčitosti některých vstupních dat (cena tepla a elektřiny, uvažování zatím nepostavených zařízení na zpracování odpadu s neznámou zpracovatelskou kapacitou) se simulují tisíce scénářů.

Z výstupů nástroje NERUDA bychom chtěli získat představu o ekonomických dopadech na producenty odpadu. K tomu je třeba mít informaci, kde producent odpad zpracoval a jakým způsobem ho přepravil. V základní verzi nástroje NERUDA však máme k dispozici pouze nejkratší hrany zajišťující přepravu mezi sousedními uzly, tím pádem nemůžeme určit, kde producent nechal zpracoval svůj odpad.

Vzniklý problém můžeme ilustrovat na příkladě (obr. 1.1). Producenti (uzly A, B, C) produkují odpad, který je svezen do uzlu D po hranách A-D, B-D, C-D. Odtud se po hraně D-E odpad veze do uzlu E, kde se rozděluje do dvou toků ke zpracovatelům F, G (po hranách E-F, E-G). V takovém případě známe pouze poměr, v jakém se odpad v uzlu E rozdělí, není ale jednoznačně určen zpracovatel pro konkrétního producenta.



Obr. 1.1: Příklad problematického toku v síti

Proto je nutné výsledky z nástroje NERUDA zpracovat pomocí dalšího nástroje se jménem PIGEON. Ten pracuje s pořadníkem producentů, který určuje přednost producenta při rozhodování, kam svůj odpad veze. Producent se poté rozhoduje na základě nejnižší ceny, což je rozdíl oproti nástroji NERUDA, který minimalizuje celkové náklady pro všechny obce najednou. Čím později se producent odpadu rozhoduje, tím horší má výběr, protože kapacity výhodných zpracovatelů jsou již obsazeny.

Ve skutečnosti ale nejsme schopni předem určit, v jakém okamžiku začnou producenti řešit, kde bude odpad zpracován. Pořadí se tedy náhodně generuje pro dostatečné množství případů.

Výpočet v nástroji PIGEON je proveden pro jeden scénář z nástroje NERUDA. Pro každý takový scénář je potřeba vypočítat desítky až stovky scénářů v nástroji PIGEON. Vzhledem k tomu, že pomocí nástroje NERUDA jsou počítány tisíce scénářů, je z časových důvodů nereálné takto analyzovat každý scénář. Je tedy nutné vybrat pouze některé reprezentativní scénáře. [11], [12]

2 ZÁKLADNÍ POJMY

Nechť X značí množinu n objektů s p znaky. Neprázdný systém

$$\Omega = \{C_1, C_2, \ldots, C_m\} \subseteq \mathcal{P}(X)$$

nazveme rozklad množiny X, jestliže platí následující podmínky:

1. $C_i \neq \emptyset$ pro $1 \leq i \leq m$,
2. $C_i \cap C_j = \emptyset$ pro $i \neq j$,
3. $C_1 \cup C_2 \cup \cdots \cup C_m = X$.

Dále vytvoříme matici $\boldsymbol{X}$ o rozměrech $n \times p$ tak, že i-tý řádek odpovídá i-tému objektu a j-tý sloupec odpovídá j-tému znaku. Tuto matici nazveme datová matice

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_{n1} & \cdots & \cdots & x_{np} \end{pmatrix}.$$

Jestliže jednotlivé znaky odpovídají číselným hodnotám, můžeme na objekty pohlížet jako na body v p-rozměrném Euklidovském prostoru $\mathbb{E}_p$.

Metrikou (vzdáleností) nazveme zobrazení $\rho : \mathbb{E}_p \times \mathbb{E}_p \to \mathbb{R}$ splňující tyto podmínky pro každé $r, s, t \in \mathbb{E}_p$:

1. $\rho(r, s) \geq 0$ (nezápornost)
2. $\rho(r, s) = 0 \Leftrightarrow r = s$
3. $\rho(r, s) = \rho(s, r)$ (symetrie)
4. $\rho(r, t) \leq \rho(r, s) + \rho(s, t)$ (trojúhelníková nerovnost)

Tímto způsobem můžeme měřit vzdálenost mezi objekty.
Jako příklad můžeme uvést Minkowského metriku, která má tvar

$$d_q(x_j, x_k) = \left[\sum_{i=1}^{p} |x_{ji} - x_{ki}|^q \right]^{1/q}, \tag{2.1}$$

kde $q \geq 1$. Na obrázku 2.1 najdeme jednotkové sféry pro nejběžnější volby q.

Abychom předešli dominanci některých znaků, je možné provést jejich standardizaci. Nechť $\boldsymbol{Z}$ je datová matice s rozměry $n \times p$. Pro všechny sloupce (znaky) z_j vypočteme střední hodnotu $\bar{z}_j$ a směrodatnou odchylku s_j podle vzorců

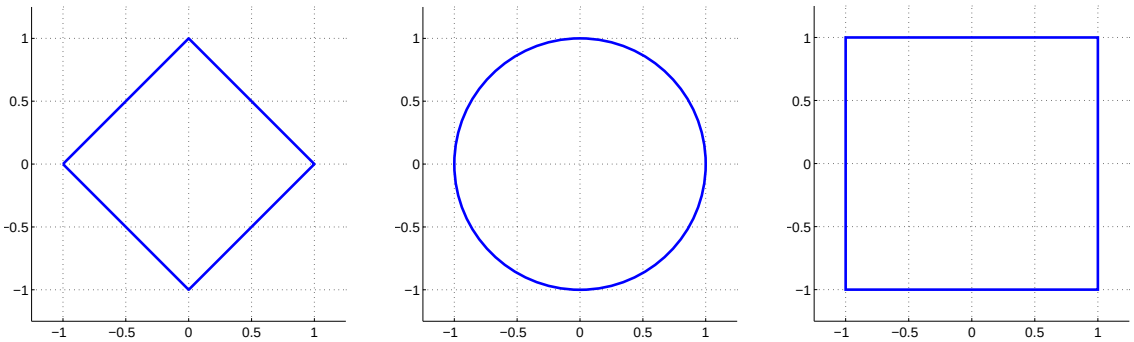
$$\bar{z}_j = \frac{1}{n} \sum_{i=1}^{n} z_{ij}, \tag{2.2}$$

$$s_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_{ij} - \bar{z}_j)^2}. \tag{2.3}$$

Poté původní hodnoty všech znaků transformujeme na standardizované hodnoty

$$x_{ij} = \frac{z_{ij} - \bar{z}_j}{s_j}. \tag{2.4}$$

Dostáváme novou datovou matici $\boldsymbol{X}$, kde jednotlivé znaky mají střední hodnotu rovnu 0 a rozptyl roven 1.



(a) Manhattanská metrika d_1 (b) Euklidovská metrika d_2 (c) Čebyševova metrika d_∞

Obr. 2.1: Příklady jednotkových sfér v různých metrikách

3 SHLUKOVÁ ANALÝZA

Pojem shluková analýza v sobě zahrnuje celou řadu metod a algoritmů, jejichž cílem je v dané množině objektů nalézt její podmnožiny (shluky) tak, aby se prvky ve shluku vzájemně podobaly a zároveň se odlišovaly od objektů mimo něj.

Algoritmy shlukové analýzy můžeme rozdělit na hierarchické a nehierarchické.

3.1 Hierarchické shlukování

Hierarchické shlukování generuje posloupnost vnořených rozkladů. Průnikem libovolných dvou shluků je tedy buď jeden z nich, nebo prázdná množina. Vytváří se stromová struktura, kterou znázorňujeme pomocí dendrogramu. Podle směru postupu při shlukování dělíme hierarchické algoritmy na aglomerativní a divizní.

3.1.1 Aglomerativní metody

U aglomerativního přístupu začínáme rozkladem Ω_0, kde každý objekt množiny X tvoří samostatný shluk a končíme rozkladem Ω_{n-1}, kde všechny prvky tvoří jediný shluk.

Postup je jednoduchý. V každém kroku shlukování vybereme dva shluky s nejmenším koeficientem nepodobnosti (viz kapitola 3.1.4). To opakujeme tak dlouho, dokud nezůstane jediný shluk obsahující všechny objekty.

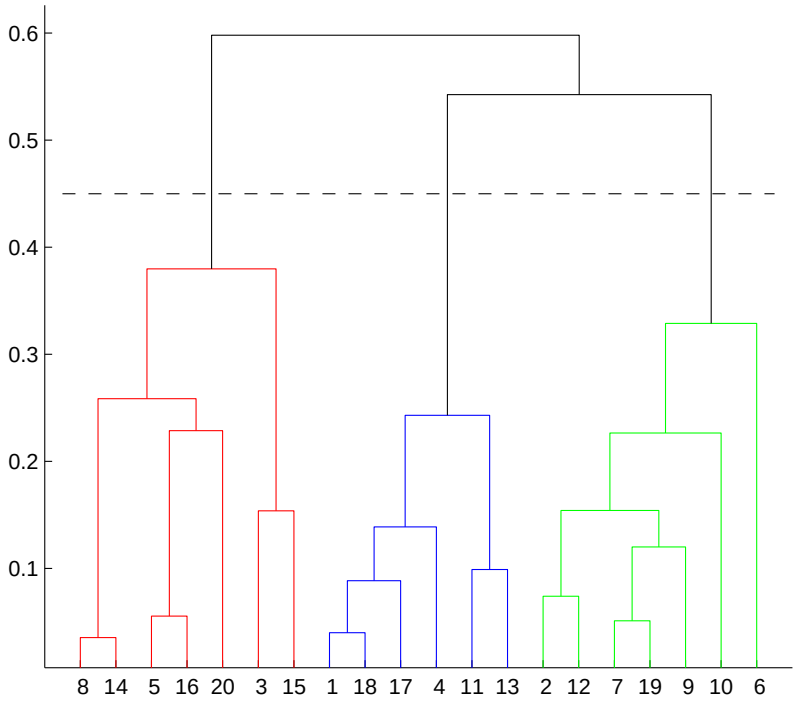
Algoritmus 1 Obecný algoritmus aglomerativního shlukování

```
1: Spočítat matici nepodobnosti shluků
2: repeat
3:     Sloučit dva shluky s nejmenším koeficientem nepodobnosti
4:     Aktualizovat matici nepodobnosti
5: until Zbývá jediný shluk
```

3.1.2 Dendrogram

Dendrogram je grafická reprezentace hierarchického shlukování. Vodorovný řez dendrogramem určuje konkrétní rozklad a uzly reprezentují jednotlivé shluky. Svislá osa znázorňuje vzdálenost (koeficient nepodobnosti) shluků, které se v dané hodnotě spojily, a na vodorovnou osu jsou vyneseny indexy objektů.

Obrázek 3.1 představuje dendrogram pro 20 objektů. Vodorovným řezem ve vzdálenosti 0,45 byly získány tři shluky, které jsou v dendrogramu barevně odlišeny.



Obr. 3.1: Dendrogram

3.1.3 Divizní metody

Zatímco u aglomerativního přístupu je v prvním kroku potřeba spočítat $n(n-1)/2$ vzdáleností, počet způsobů, jak rozdělit shluk na dvě části, je $2^{n-1}-1$. To je neporovnatelně více, obzvláště pro velká n, proto se v této práci těmito algoritmy nebudeme zabývat.

3.1.4 Koeficient nepodobnosti shluků

Nechť ρ je libovolná metrika a $C_i, C_j \in \Omega$ jsou shluky v rozkladu Ω. Zobrazení $d : \Omega \times \Omega \to \mathbb{R}$ nazveme koeficientem nepodobnosti shluků (nebo také vzdáleností shluků), jestliže splňuje tyto podmínky:

1. $d(C_i, C_i) = 0$,
2. $d(C_i, C_j) \geq 0$,
3. $d(C_i, C_j) = d(C_j, C_i)$.

Matici $\boldsymbol{D} = (d_{ij})$, kde $d_{ij} = d(C_i, C_j), 1 \leq i, j \leq |\Omega|$, nazveme matice nepodobnosti (shluků), případně matice vzdáleností (shluků).

Nyní si uvedeme některé běžné způsoby zavedení koeficientu nepodobnosti shluků.

Metoda nejbližšího souseda

Nechť ρ je libovolná vzdálenost a A, B jsou shluky rozkladu Ω. Metoda nejbližšího souseda definuje koeficient nepodobnosti shluků A a B takto:

$$d(A,B) = \min_{\substack{x_i \in A \\ x_j \in B}} \{\rho(x_i, x_j)\}\,. \tag{3.1}$$

Vzdáleností shluků A, B tedy rozumíme vzdálenost dvou nejbližších prvků shluku A a shluku B.

Metoda nejvzdálenější souseda

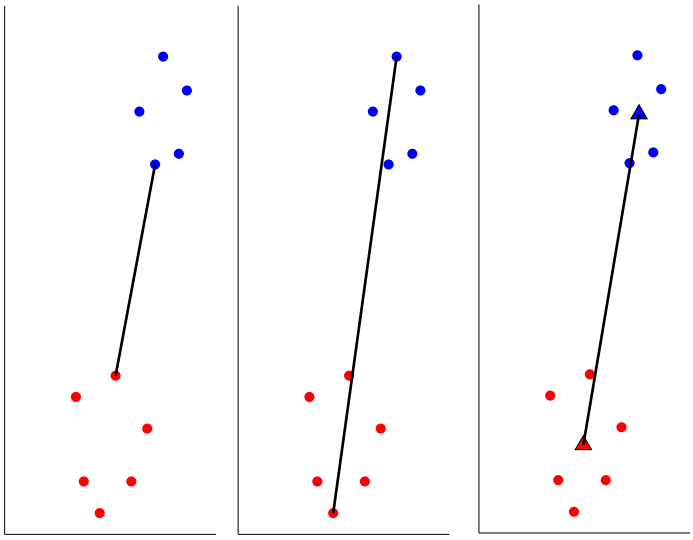
Jedná se o obdobu metody nejbližšího souseda, v tomto případě však uvažujeme vzdálenost dvou nejvzdálenějších prvků

$$d(A,B) = \max_{\substack{x_i \in A \\ x_j \in B}} \{\rho(x_i, x_j)\}\,. \tag{3.2}$$

Centroidní metoda

U centroidní metody definujeme vzdálenost dvou shluků jako vzdálenost jejich centroidů (těžišť)

$$d(A,B) = \rho(\bar{x}_A, \bar{x}_B), \quad \text{kde } \bar{x}_A = \frac{1}{|A|}\sum_{x_i \in A} x_i, \quad \bar{x}_B = \frac{1}{|B|}\sum_{x_j \in B} x_j. \tag{3.3}$$



Obr. 3.2: Metoda nejbližšího souseda / Metoda nejvzdálenějšího souseda / Centroidní metoda

Metoda průměrné vazby

Tato metoda bere za vzdálenost dvou shluků průměr vzdáleností všech dvojic prvků z různých shluků, tedy

$$d(A,B) = \frac{1}{|A||B|} \sum_{x_i \in A} \sum_{x_j \in B} \rho(x_i, x_j). \tag{3.4}$$

Mezi další používané metody patří *mediánová metoda, vážená metoda průměrné vazby* nebo *Wardova metoda*, jejichž definice a vlastnosti najdeme v [5] a [8].

3.1.5 Lance-Williamsův vzorec

V předchozích odstavcích jsme si uvedli některé explicitní vzorce pro koeficienty nepodobnosti shluků. Pro praktický výpočet je však výhodnější použít tento způsob výpočtu:

1. $d\left(\{x_i\}, \{x_j\}\right) = \rho(x_i, x_j)$
2. Nechť $T = R \cup S$ je shluk rozkladu Ω_{i+1} získaný sjednocením shluků $R, S \in \Omega_i$. Potom pro všechny shluky K i-tého rozkladu, které přejdou beze změny do rozkladu Ω_{i+1}, platí

$$d(K,T) = \alpha_r d(K,R) + \alpha_s d(K,S) + \beta d(R,S) + \gamma \left| d(K,R) - d(K,S) \right|. \tag{3.5}$$

Tímto způsobem můžeme aktualizovat matici nepodobností shluků, aniž bychom potřebovali původní data. Koeficienty $\alpha_r, \alpha_s, \beta$ a γ pro většinu běžně používaných metod najdeme v tabulce 3.1.

Tab. 3.1: Koeficienty pro Lance-Williamsův vzorec [6]

Shlukovací metoda	α_r	α_s	β	γ
Metoda nejbližšího souseda	$1/2$	$1/2$	0	$-1/2$
Metoda nejvzdálenějšího souseda	$1/2$	$1/2$	0	$1/2$
Cetroidní metoda	$\frac{\lvert R\rvert}{\lvert S\rvert+\lvert R\rvert}$	$\frac{\lvert S\rvert}{\lvert S\rvert+\lvert R\rvert}$	$\frac{-\lvert R\rvert\lvert S\rvert}{(\lvert S\rvert+\lvert R\rvert)^2}$	0
Mediánová metoda	$1/2$	$1/2$	$-1/4$	0
Metoda průměrné vazby (nevážená)	$\frac{\lvert R\rvert}{\lvert S\rvert+\lvert R\rvert}$	$\frac{\lvert S\rvert}{\lvert S\rvert+\lvert R\rvert}$	0	0
Metoda průměrné vazby (vážená)	$1/2$	$1/2$	0	0
Wardova metoda	$\frac{\lvert R\rvert+\lvert K\rvert}{\lvert S\rvert+\lvert R\rvert+\lvert K\rvert}$	$\frac{\lvert S\rvert+\lvert K\rvert}{\lvert S\rvert+\lvert R\rvert+\lvert K\rvert}$	$\frac{-\lvert K\rvert}{\lvert S\rvert+\lvert R\rvert+\lvert K\rvert}$	0

Je třeba dodat, že pro odvození koeficientů centroidní, mediánové a Wardovy metody byl použit čtverec Euklidovské vzdálenosti. Pro jinou volbu vzdálenosti tedy obecně dostáváme jinou metodu.

3.1.6 Ilustrační příklad

Máme zadaných pět bodů v rovině:

$$A_1 = (7,7), A_2 = (1,1), A_3 = (1,2), A_4 = (7,5), A_5 = (3,3).$$

Tyto body (dále je budeme značit jen jejich indexem) chceme shlukovat pomocí centroidní metody s euklidovskou vzdáleností a použijeme k tomu explicitní vzorec. V prvním kroku spočítáme matici vzdáleností $\boldsymbol{D} = (d_{ij})$.

$$\boldsymbol{X}^{(0)} = \begin{pmatrix} 7 & 7 \\ 1 & 1 \\ 1 & 2 \\ 7 & 5 \\ 3 & 3 \end{pmatrix} \begin{matrix} \{1\} \\ \{2\} \\ \{3\} \\ \{4\} \\ \{5\} \end{matrix} \qquad \boldsymbol{D}^{(0)} = \begin{pmatrix} 0 & & & & \\ 8{,}49 & 0 & & & \\ 7{,}81 & \mathbf{1} & 0 & & \\ 2 & 7{,}21 & 6{,}71 & 0 & \\ 5{,}66 & 2{,}83 & 2{,}24 & 4{,}47 & 0 \end{pmatrix}$$

Nejmenší nenulový prvek matice vzdálenosti je d_{32}, proto sloučíme shluky $\{2\}$ a $\{3\}$ (obr. 3.3a). Vytvoří se tak shluk $\{2,3\}$, který bude charakterizován těžištěm objektů 2 a 3. Poté přepočítáme matici vzdáleností.

$$\boldsymbol{X}^{(1)} = \begin{pmatrix} 7 & 7 \\ 7 & 5 \\ 3 & 3 \\ 1 & 1{,}5 \end{pmatrix} \begin{matrix} \{1\} \\ \{4\} \\ \{5\} \\ \{2,3\} \end{matrix} \qquad \boldsymbol{D}^{(1)} = \begin{pmatrix} 0 & & & \\ \mathbf{2} & 0 & & \\ 5{,}66 & 4{,}47 & 0 & \\ 8{,}14 & 6{,}95 & 2{,}5 & 0 \end{pmatrix}$$

Opět vybereme nejmenší nenulový prvek, v tomto případě d_{21}. Pozicím $1, 2$ odpovídají shluky $\{1\}$, $\{4\}$, které sloučíme (obr. 3.3b).

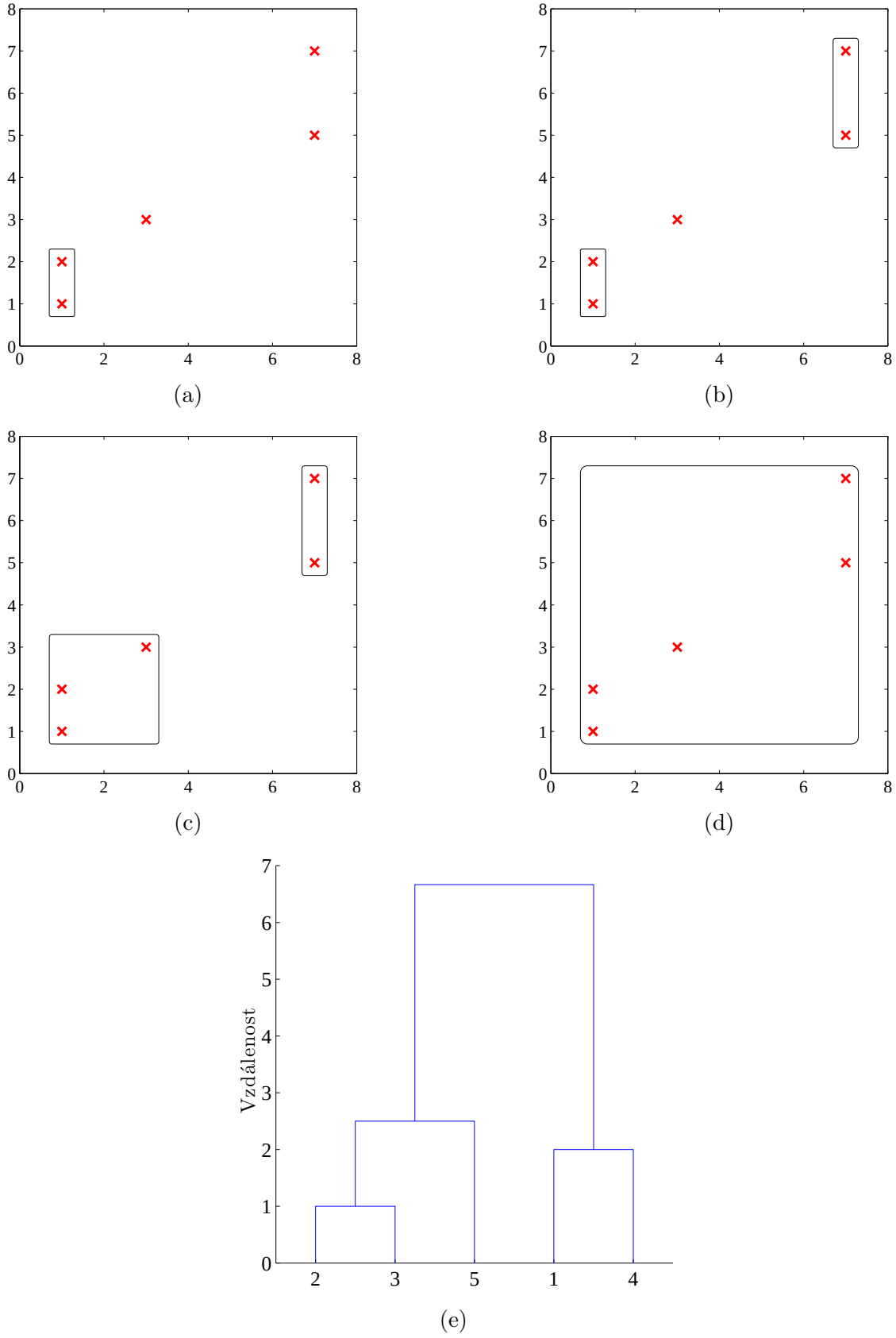
$$\boldsymbol{X}^{(2)} = \begin{pmatrix} 3 & 3 \\ 1 & 1{,}5 \\ 7 & 6 \end{pmatrix} \begin{matrix} \{5\} \\ \{2,3\} \\ \{1,4\} \end{matrix} \qquad \boldsymbol{D}^{(2)} = \begin{pmatrix} 0 & & \\ \mathbf{2{,}5} & 0 & \\ 5 & 7{,}5 & 0 \end{pmatrix}$$

Opakováním výše zmíněného postupu ze shluků $\{5\}$ a $\{2,3\}$ dostáváme nový shluk $\{2,3,5\}$ (obr. 3.3c), který reprezentujeme těžištěm bodů $2, 3$ a 5.

$$\boldsymbol{X}^{(3)} = \begin{pmatrix} 7 & 6 \\ 1{,}67 & 2 \end{pmatrix} \begin{matrix} \{1,4\} \\ \{2,3,5\} \end{matrix} \qquad \boldsymbol{D}^{(3)} = \begin{pmatrix} 0 & \\ \mathbf{6{,}67} & 0 \end{pmatrix}$$

V tomto kroku obdržíme 1 shluk, ve kterém jsou všechny objekty (obr. 3.3d), algoritmus tedy ukončíme.

Celý postup včetně vzdáleností mezi jednotlivými shluky lze graficky zobrazit pomocí dendrogramu (obr. 3.3e).



Obr. 3.3: Algoritmus centroidní metody

3.2 Nehierarchické shlukování

Na rozdíl od hierarchických metod se v těchto metodách mohou prvky přesouvat mezi shluky. Většinou je třeba začít vytvořením počátečního rozkladu, který se poté iteračně upravuje přeskupováním prvků mezi shluky tak, aby se minimalizovala vhodně zvolená účelová funkce.

3.2.1 Algoritmus k-means

Algoritmus k-means spočívá v minimalizaci sumy čtverců vzdáleností mezi prvky a těžišti shluků, do kterých patří. To můžeme zapsat jako funkcionál

$$J(C) = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left[d_2 \left(x_i^{(j)}, c_j \right) \right]^2, \tag{3.6}$$

kde d_2 značí Euklidovskou metriku, k předem zvolený počet shluků, n_j počet prvků v j-tém shluku, c_j těžiště (centroid) j-tého shluku a $x_i^{(j)}$ je i-tý prvek j-tého shluku.

Shlukování pomocí algoritmu k-means probíhá následujícím způsobem.

Algoritmus 2 K-Means

1: Vytvoření počátečního rozkladu
2: Výpočet centroidů všech shluků
3: **repeat**
4: Přiřazení každého prvku do shluku, k jehož centroidu má nejmenší vzdálenost
5: Přepočet souřadnic centroidů
6: **until** Nedojde k přesunu žádného prvku

Lze dokázat, že metoda konverguje v konečném počtu kroků k lokálnímu minimu funkcionálu J. Těchto minim může obecně existovat více v závislosti na počátečním rozkladu. Proto v praxi algoritmus použijeme několikrát s různými počátečními rozklady a vybereme ten nejlepší rozklad z hlediska hodnoty účelové funkce. Ani tak bohužel není zaručeno nalezení globálního minima.

Počáteční rozklad

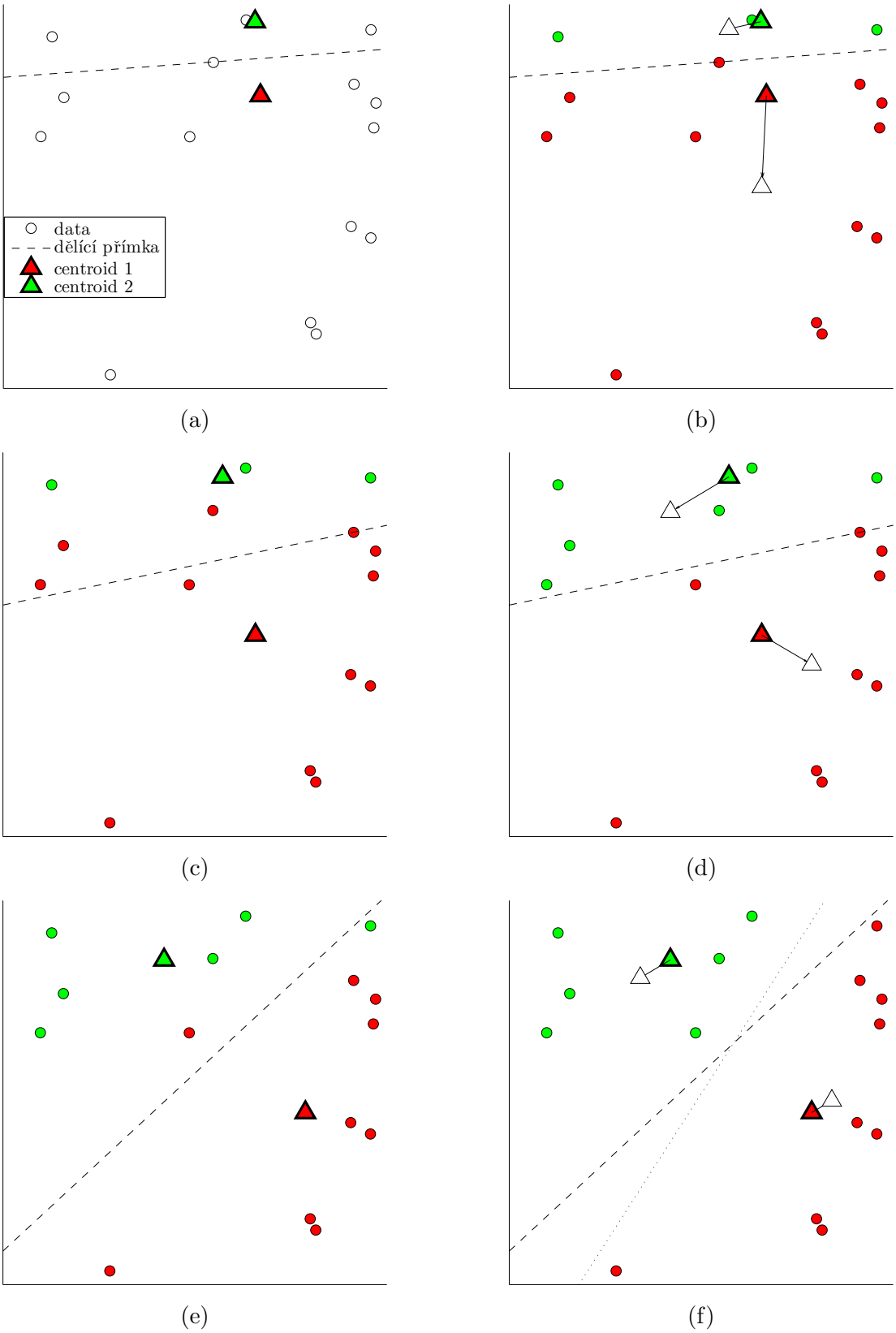
Zabývejme se nyní volbou počátečního rozkladu. Nejběžnějším způsobem je náhodné vygenerování k bodů v prostoru $\mathbb{E}_p$. Každý prvek poté přiřadíme k jeho nejbližšímu centroidu, čímž obdržíme počáteční rozklad. Nevýhodou tohoto přístupu je, že můžeme v počátečním rozkladu obdržet prázdný shluk. Snadno se jí však zbavíme, pokud počáteční body budeme náhodně vybírat z množiny dat.

Existuje mnoho dalších heuristik zabývajících se vhodnou volbou počátečního rozkladu. Tyto metody však přesahují rámec tohoto textu a nebudeme se jimi zabývat.

Ilustrační příklad

V tomto odstavci je na příkladu ukázán princip algoritmu k-means. Úkolem je rozdělit 15 dvourozměrných objektů z obrázku 3.4a do dvou shluků.

V prvním kroku se náhodně vygenerují dva centroidy (obr. 3.4a). Poté se data rozdělí do shluků tak, aby euklidovská vzdálenost mezi daty a centroidy byla minimální. Nové centroidy se určí jako těžiště bodů ve shluku (obr. 3.4b). Opět se minimalizují vzdálenosti mezi daty a novými centroidy, čímž dojde k přesunu některých objektů mezi shluky, a vypočítají se nové souřadnice centroidů (obr. 3.4c a 3.4d). V další iteraci ještě dojde k přesunu objektů (obr. 3.4e), nové centroidy ale shluky nezmění, proto je algoritmus ukončen (obr. 3.4f).



Obr. 3.4: Algoritmus k-means

3.2.2 Algoritmus fuzzy c-means

Doposud zmíněné algoritmy každý prvek jednoznačně přiřadily do určitého shluku, a to i v případě, že ležel na „hranici“ mezi dvěma shluky. Výjimkou nebyly ani odlehlé hodnoty (outliers), které například u algoritmu k-means mohly vychýlit pozici centroidů. Tyto nedostatky se pokouší řešit algoritmus fuzzy c-means. Jedná se o obdobu algoritmu k-means, v tomto případě však každému prvku i přiřadíme příslušnost u_{ij} ke všem shlukům j. [3]

Minimalizuje se funkcionál

$$J(U, C) = \sum_{j=1}^{k} \sum_{i=1}^{n} (u_{ij})^m \left[d_2 \left(x_i, c_j \right) \right]^2, \tag{3.7}$$

kde m je konstanta určující „rozostřenost“ shluků. Pro zaručení konvergence k lokálnímu minimu zmíněného funkcionálu volíme m z intervalu $(1, \infty)$. Tohoto optima ovšem nedosáhneme v konečnému počtu kroků, proto algoritmus ukončíme v momentě, kdy se účelová funkce změní o menší hodnotu než zvolené ε.

Algoritmus 3 Fuzzy c-means

1: Výběr / vygenerování k počátečních centroidů
2: **repeat**
3: Výpočet příslušnosti bodů k jednotlivým centroidům
4: Výpočet nových centroidů na základě příslušnosti bodů
5: **until** $J^{i-1} - J^i < \varepsilon$

Jednotlivé příslušnosti spočítáme pomocí vzorce

$$u_{ij} = \frac{1}{\sum_{l=1}^{k} \left[\frac{d_2(i,c_j)}{d_2(i,c_l)} \right]^{\frac{2}{m-1}}}, \qquad 1 \leq i \leq n,\ 1 \leq j \leq k. \tag{3.8}$$

Pro jejich hodnoty platí

$$u_{ij} \in \langle 0, 1 \rangle, \quad \sum_{l=1}^{k} u_{il} = 1. \tag{3.9}$$

Centroidy shluků poté přepočítáme jako

$$c_j = \frac{\sum_{i=1}^{n} (u_{ij})^m x_i}{\sum_{i=1}^{n} (u_{ij})^m}, \qquad 1 \leq j \leq k. \tag{3.10}$$

Konstanta m výrazně ovlivňuje výsledky shlukování. Pro $m \to \infty$ mají všechny prvky stejnou příslušnost ke všem shlukům, zatímco pro $m \to 1$ přechází metoda v k-means.

Volbě optimální konstanty m se věnují Schwammle a Jensen v [10]. Podle nich závisí na dimenzi p a počtu prvků n vztahem

$$m = 1 + \left(\frac{1418}{n} + 22{,}05\right) p^{-2} + \left(\frac{12{,}33}{n} + 0{,}243\right) p^{-0{,}0406\ln(n)-0{,}1134}. \tag{3.11}$$

3.3 Validace shluků

Výše zmíněné algoritmy provedou rozklad množiny na shluky bez ohledu na skutečnou strukturu dat. Proto vzniká potřeba analyzovat, jak kompaktní nalezené shluky jsou. Takovému postupu říkáme validace shluků. Jedním ze způsobů, jak ji provést, je využití takzvaných siluet, jejichž konstrukci a aplikaci si nyní předvedeme. [9]

3.3.1 Konstrukce siluet

Nechť i je i-tý objekt patřící shluku A[1]. Pokud shluk A obsahuje nějaké další prvky, můžeme vypočítat průměrnou vzdálenost mezi prvkem i a všemi ostatními prvky j shluku A jako

$$a(i) = \frac{\sum_{j \in A} \rho(i,j)}{|A| - 1} \qquad i \neq j. \tag{3.12}$$

Vzdálenost ρ může být libovolná, my se však omezíme na $\rho = d_2$, abychom mohli vzniklé siluety vzájemně porovnávat. Uvažujme dále jiný shluk $C \neq A$. Vypočítáme průměrnou vzdálenost mezi prvkem i a všemi prvky shluku C

$$\bar{\rho}(i,C) = \frac{\sum_{j \in C} \rho(i,j)}{|C|}. \tag{3.13}$$

Tuto hodnotu vypočítáme pro všechny takové shluky C a definujeme

$$b(i) = \min_{C \neq A} \bar{\rho}(i,C). \tag{3.14}$$

Nyní zkonstruujeme siluetu prvku i

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{pro } a(i) < b(i) \\ 0 & \text{pro } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{pro } a(i) > b(i), \end{cases} \tag{3.15}$$

což lze ekvivalentně zapsat jako

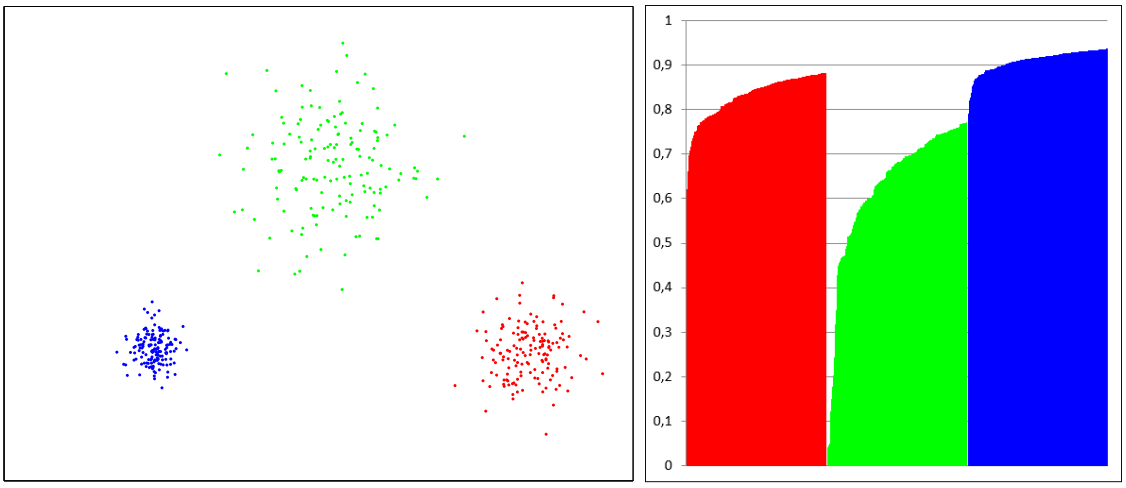
$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}. \tag{3.16}$$

1. Abychom mohli tento postup použít i pro algoritmus fuzzy c-means, musíme přiřadit prvek i do shluku, ke kterému má největší příslušnost. Častěji se však pro tento algoritmus používají postupy, které zohledňují i příslušnosti u_{ij}, zabývat se jimi ale nebudeme.

Pokud shluk A obsahuje jediný prvek i, definujeme $s(i) = 0$. Z výše uvedených definic je zřejmé, že $s(i) \in \langle -1, 1 \rangle$.

3.3.2 Grafická reprezentace siluet

Siluety prvků i jednotlivých shluků A lze zobrazit do dvourozměrného grafu, kde na svislou osu nanášíme hodnoty $s(i)$ a na vodorovnou osu podle hodnoty $s(i)$ seřadíme prvky shluku A. Mluvíme poté o siluetě shluku A. Pokud do grafu naskládáme siluety všech shluků, dostaneme siluetu rozkladu (obr. 3.5b).



(a) Shlukovaná data (b) Silueta rozkladu

Obr. 3.5: Ukázka rozkladu a jeho siluety

3.3.3 Interpretace siluet

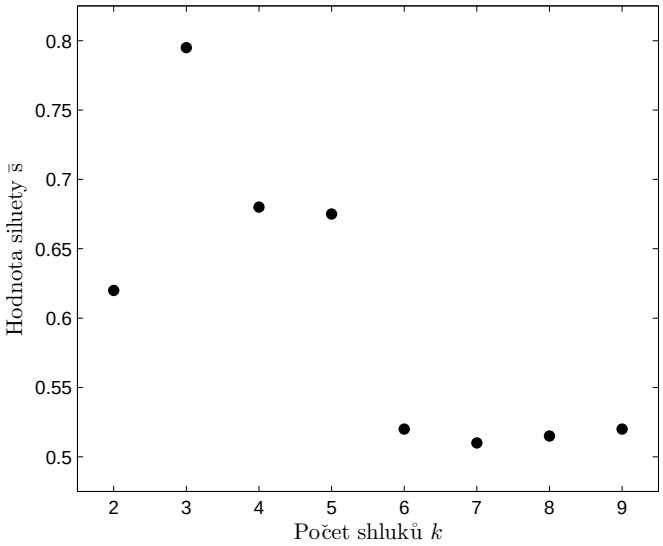
Věnujme se nyní interpretaci možných hodnot siluet. Představme si nejprve, že naše data tvoří shluky malé velikosti, které jsou od sebe daleko vzdálené. Pro prvek i potom zřejmě bude platit, že hodnota $a(i)$ (rovnice 3.12) je o hodně menší, než hodnota $b(i)$ (rovnice 3.14). Čím blíže k sobě budou prvky v rámci jednotlivých shluků a čím dále od sebe tyto shluky budou, tím menší bude poměr $a(i)/b(i)$. Z rovnice 3.15 poté plyne, že pro takovýto případ se hodnota $s(i)$ blíží jedné.

Takový případ nastal pro prvky modrého shluku z obrázku 3.5a. Vysoké hodnoty $s(i)$ všech jeho prvků potvrzují, že se jedná o dobře oddělený a kompaktní shluk. Prvky zeleného shluku jsou mnohem více rozptýleny, proto je jeho silueta níž.

3.3.4 Určení optimálního počtu shluků pomocí siluet

V předchozím odstavci jsme ukázali, že velikost siluet značí, jak dobře je prvek ve shluku zařazen. Obecně hledáme rozklad takový, že většina prvků je dobře zařazena. V takovém případě očekáváme vysokou hodnotu siluety rozkladu $\bar{s}$, kterou definujeme jako průměr hodnot $s(i)$ pro všechny prvky i rozkladu. Toho můžeme využít při určování optimálního počtu shluků. Provedeme rozklady pro různé počty shluků k, pro každý rozklad napočítáme $\bar{s}$ a vybereme největší hodnotu. Odpovídající počet shluků prohlásíme za optimální.

Aplikujme nyní tento postup na data z obrázku 3.5a. Algoritmem k-means rozdělíme data postupně do dvou až devíti shluků. Napočítané hodnoty vidíme v grafu na obrázku 3.6. Nejlepší siluety rozkladu $\bar{s}$ jsme dosáhli pro 3 shluky, což odpovídá i našemu předpokladu na základě obrázku.



Obr. 3.6: Určení optimálního počtu shluků

3.4 Poznámka o shlukování ve vyšších dimenzích

V prostorech vyšších dimenzí se běžně používaná euklidovská metrika chová jinak, než jak ji známe z prostorů dimenzí 2 a 3. Tento jev se nazývá prokletí dimenzionality. V některých případech se projevuje tak, že se rozdíl vzdáleností nejbližšího a nejvzdálenějšího prvku k danému objektu blíží nule. Euklidovská vzdálenost tedy ztrácí schopnost popsat kontrast mezi daty. Podle [1] se v těchto případech více hodí

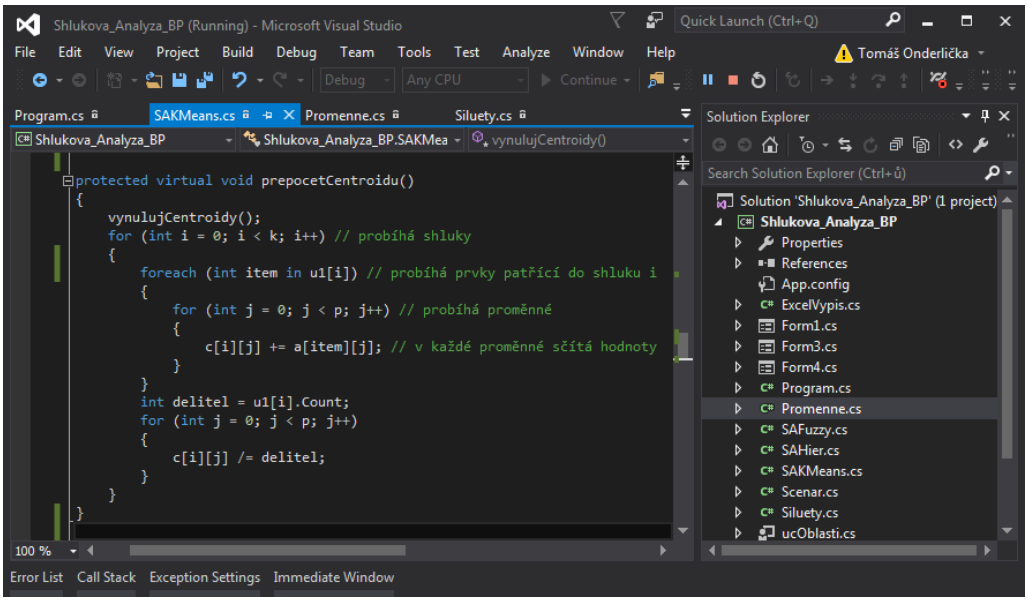
d_1 metrika. Ještě lepších výsledků však dosahují takzvané zlomkové metriky[2] (fractional distance metrics). Jedná se o obdobu Minkowského metriky (rovnice 2.1), avšak v tomto případě dovolíme $q \in (0, 1)$. Jejich implementací se však zabývat nebudeme.

2. Název metrika je zde nepřesný. Z matematického hlediska se o metriku nejedná, protože nesplňuje trojúhelníkovou nerovnost.

4 VYTVOŘENÝ PROGRAM

V rámci této práce vznikl software, který dokáže scénáře vygenerované nástrojem NERUDA zpracovat a poté je shlukovat pomocí algoritmů, které byly představeny v teoretické části této práce. Výsledky shlukování včetně siluet a jejich grafické reprezentace poté exportuje do sešitu programu MS Excel.

Software vznikl ve vývojovém prostředí Visual Studio od firmy Microsoft a je naprogramován v jazyce C#.



Obr. 4.1: Ukázka prostředí Microsoft Visual Studio 2015

4.1 Postup při shlukování

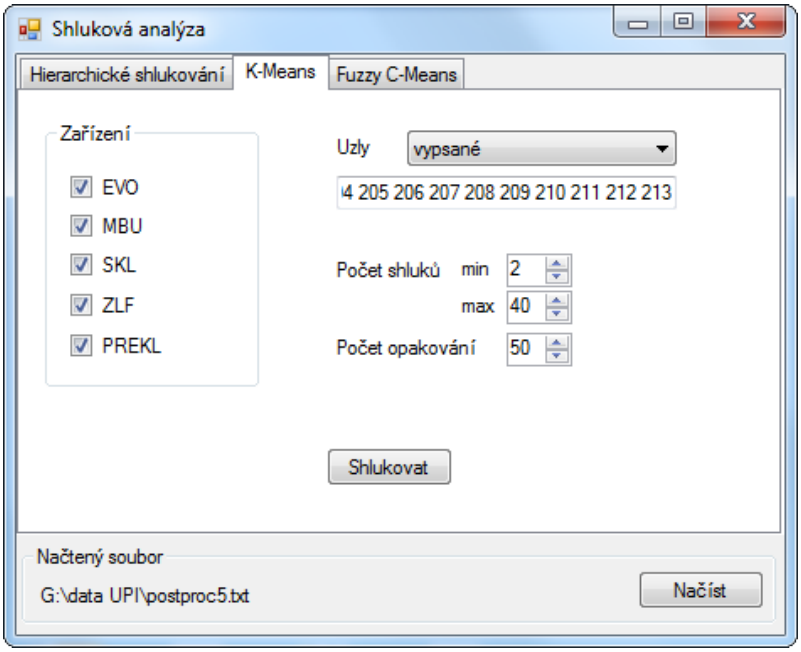
Uživatel nejprve načte soubor se scénáři, které se uloží do paměti. Poté vybere typ shlukování a nastaví jeho základní parametry. Ze zadaných údajů program vybere proměnné (znaky) scénářů. Ty poté následujícím způsobem zpracuje. Nejprve se odstraní proměnné s nulovým rozptylem[1] a proměnné, které jsou identickými kopiemi jiných proměnných[2]. Následuje standardizace dat podle vzorce 2.4. Z takto vytvořených proměnných se generuje datová matice, nad níž proběhne shlukování podle zvoleného nastavení.

1. Proměnná s nulovým rozptylem má stejné hodnoty pro všechny scénáře, nemá tedy na shlukování žádný vliv.
2. Proměnná, která má pro všechny scénáře stejné hodnoty jako některá jiná proměnná. Jejím vynecháním není z hlediska shlukování ztracena žádná informace.

4.2 Možnosti nastavení a uživatelské rozhraní

Nastavení shlukování se provádí pomocí jednoduchého grafického uživatelské rozhraní. K načtení souboru s daty slouží tlačítko `Načíst` v pravé dolní části okna. V horním panelu potom vybíráme typ algoritmu shlukování. Na výběr jsou metody uvedené v této práci, tedy

- `Hierarchické shlukování`,
- `K-Means`,
- `Fuzzy C-Means`.



Obr. 4.2: Ukázka uživatelského rozhraní

U všech algoritmů má uživatel možnost zadat interval počtu shluků k, které chce získat. Spodní hodnotu zadává položkou `min`, horní hodnotu položkou `max`. Pro algoritmy hierarchického shlukování má tento interval vliv pouze na to, pro jaké počty shluků se budou počítat siluety.

Pro hierarchické shlukování je nutné zvolit koeficient nepodobnosti shluků. K tomu slouží rozbalovací seznam s názvem `Metody` a možné volby jsou

- `Nejbližší soused`,
- `Nejvzdálenější soused`,
- `Centroidní metoda`,
- `Mediánová metoda`,
- `Průměrná vazba (nevážená)`,
- `Průměrná vazba (vážená)`.

Výpočet pak probíhá pomocí Lance-Williamsova vzorce (kapitola 3.1.5). Ze vzdáleností je na výběr

- `Euklidovská`,
- `Čtverec euklidovské`,
- `Manhattan`,
- `Čebyševova`.

U zbylých dvou algoritmů volíme `Počet opakování`, čímž specifikujeme, kolik různých počátečních rozkladů pro každé k bude program generovat. Ze všech vypočtených rozkladů se poté vybere ten s nejmenší hodnotou účelové funkce. Pro fuzzy c-means přibývá ještě volba konstanty m. Pokud zadáme číslo 0, program její hodnotu vypočte podle vzorce 3.11.

Volby `Zařízení` a `Uzly` ovlivňují proměnné, které budou vstupovat do výpočtu. Jejich význam je vysvětlen v kapitole 5.1.

4.3 Výstup

Výstupem shlukování je sešit v programu MS Excel. Ten se generuje pro zvolené hodnoty počtu shluků a ukládá se automaticky do složky s aplikací. Nachází se v něm přehled nastavených parametrů shlukování, datová matice, výpis shluků a jejich prvků. Pro interpretaci výsledků je však nejdůležitější list s názvem `Siluety`. Výřez z něj můžeme vidět na obrázku 4.3. Za zmínku stojí druhý sloupec, ve kterém se nachází číslo „sousedního“ shluku jednotlivých prvků. Jedná se o shluk, pro který byla napočítána minimální vzdálenost $\bar{\rho}(i, C)$ (rovnice 3.13). V případě nízké hodnoty siluety $s(i)$ tak máme představu, kam by prvek alternativně mohl patřit.

	A	B	C	D
1	**Shluk**	**Soused**	**s(i)**	**Scénář**
3366	23	25	0,7163	1939
3367	23	13	0,7168	1119
3368	23	25	0,7168	4090
3369	23	25	0,718	4143
3370	23	13	0,7182	1690
3371	23	25	0,7183	3222
3372	23	25	0,7184	270
3373	23	16	0,7189	4892
3374	23	25	0,7189	4252
3375	23	25	0,719	3440
3376	23	25	0,719	573

Obr. 4.3: Část výsledků shlukování

5 APLIKAČNÍ ČÁST

5.1 Data

Představme si nejprve data, která budeme shlukovat. Jedná se o 5 000 scénářů odpadového hospodářství v ČR optimalizovaných nástrojem NERUDA. Jako proměnné uvažujeme zpracovatelskou kapacitu jednotlivých zařízení[1] a jejich cenu na bráně[2].

V datech se vyskytují tyto typy zařízení:

- `EVO` - zařízení pro energetické využití odpadu (spalovny),
- `MBÚ` - zařízení na mechanicko-biologickou úpravu,
- `SKL` - skládky,
- `ZLF` - zařízení pro zpracování lehké frakce,
- `PREKL` - překládací stanice.

Každé zařízení přísluší některému uzlu i z množiny $I = \{1, \ldots, 213\}$. Jedná se o obce s rozšířenou působností a města s více než 10 000 obyvateli.

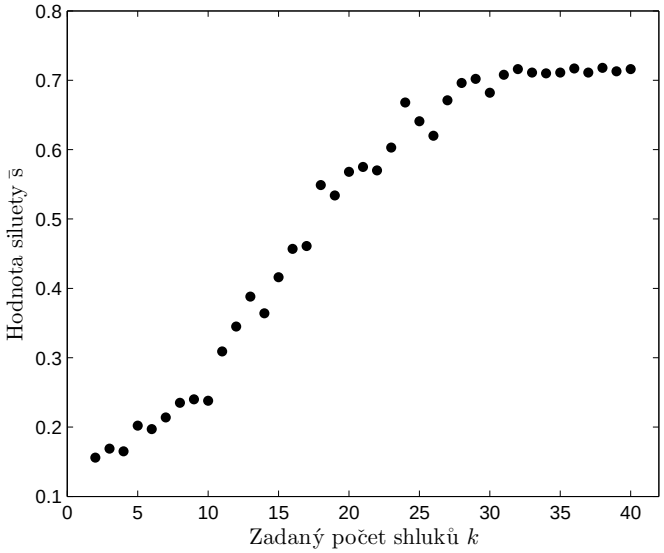
5.2 Aplikace na podoblast ČR

Nejprve budeme scénáře shlukovat pouze podle zařízení v určité podoblasti ČR. Pro tento účel zvolíme Zlínský kraj, kterému odpovídají uzly 201 až 213 a 46 proměnných. Začneme algoritmem k-means. Budeme hledat rozklady pro 2 až 40 shluků a provedeme 50 různých rozkladů pro každou hodnotu k. Z těchto 50 zvolíme nejlepší rozklad podle hodnoty účelové funkce. Obdržené výsledky poté vyhodnotíme pomocí siluet.

Vypočítané hodnoty siluet rozkladu $\bar{s}$ v závislosti na zadaném počtu shluků k vidíme v grafu na obrázku 5.1. Přibližně od $k = 28$ se hodnota $\bar{s}$ ustálí. Z hlediska siluet jsou tedy rozklady s 28 až 40 shluky podobně kvalitní. Vybereme z nich takový, který má co nejméně špatně zařazených prvků, tedy prvků i takových, že $s(i) < 0$. Tímto postupem získáme $k = 34$. Grafická interpretace siluet pro tento rozklad se nachází v příloze A.

Metody hierarchického shlukování se chovají v tomto případě o poznání hůře. Velmi brzy se spojí prvky shluků, které algoritmus k-means rozpoznal jako rozdílné. Řetězovou reakcí potom vznikne jeden shluk, který obsahuje téměř všechny objekty, a zároveň zůstane mnoho objektů, které tvoří samostatný shluk.

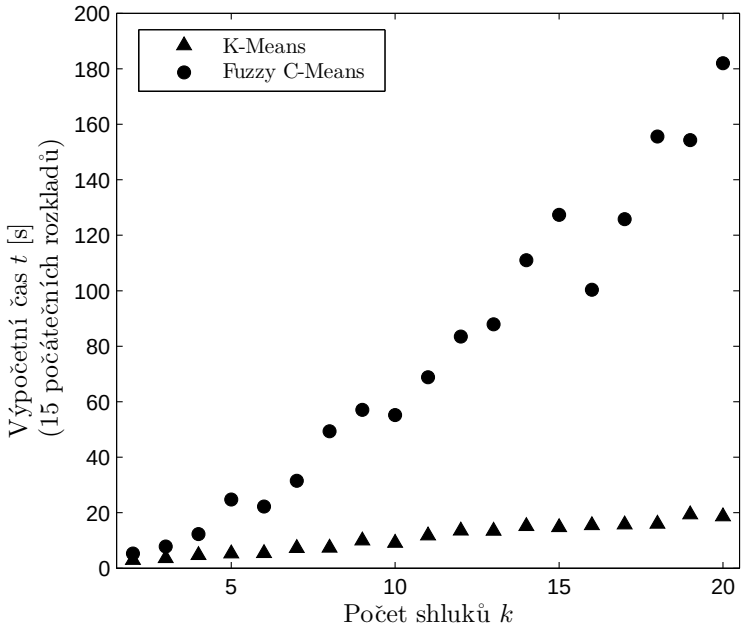
1. Zpracovatelská kapacita znamená množství odpadu v tunách, které je zařízení schopno ročně zpracovat.
2. Cena na bráně určuje poplatek za zpracování jedné tuny odpadu.



Obr. 5.1: Siluety $\bar{s}$, algoritmus k-means

Porovnání výpočetní náročnosti algoritmů k-means a fuzzy c-means pro použitá data najdeme na obrázku 5.2. Pro počty shluků 2 až 20 bylo generováno 15 počátečních rozkladů a měřila se doba t v sekundách mezi vytvořením 1. počátečního rozkladu a ukončením výpočtu pro 15. počáteční rozklad. Vidíme, že pro rostoucí k potřebuje fuzzy c-means podstatně více času než k-means. Pro naši aplikaci se proto nevyplatí.

Interpretujme nyní výsledky z algoritmu k-means pomocí obdržených siluet. Na první pohled nás zaujmou čtyři široké a vysoké siluety shluků $1, 2, 12$ a 19. Z vysokých hodnot $s(i)$ můžeme usuzovat, že jsou si všechny scénáře velmi podobné. Velká šířka značí, že se v těchto shlucích nachází mnoho scénářů. Pro analýzu nástrojem PIGEON z těchto shluků můžeme vybrat jeden scénář (s největší hodnotou siluety) a pravděpodobně dostaneme výsledky reprezentativní pro všechny prvky odpovídajících shluků. Podobně můžeme postupovat pro velké množství menších shluků s vysokými siluetami, například shluky $13, 14, 21$. V několika shlucích $(8, 15, 34$ a další) najdeme prvky se zápornými siluetami $s(i)$. To znamená, že se od ostatních prvků shluku mohou podstatně odlišovat, proto musíme být při jejich další analýze opatrní. Další možností by bylo tyto prvky přesunout do sousedních shluků a přepočítat siluety.



Obr. 5.2: Porovnání výpočtové náročnosti

5.3 Aplikace na celou ČR

Aplikujme nyní stejný postup pro všechny proměnné, kterých je nyní 581. Kromě prudkého nárůstu výpočetního času nás negativně překvapí také nízká hodnota siluet $\bar{s}$. Ta se pro všechny rozklady pohybuje těsně pod nulou. Pokud se blíže podíváme na vzdálenosti mezi prvky, zjistíme, že se všechny pohybují ve velmi úzkém intervalu. Bohužel se projevilo prokletí dimenzionality, které jsme si přiblížili v kapitole 3.4. S tímto problémem si námi představené algoritmy používající běžné vzdálenosti neporadí.

6 ZÁVĚR

V práci je nejprve stručně nastíněna problematika nástrojů NERUDA a PIGEON, ze které plyne nutnost použití shlukové analýzy.

Následuje představení základních algoritmů, mezi které patří hierarchické shlukování, algoritmus k-means a algoritmus fuzzy c-means. Poté je pozornost věnována interpretaci výsledků shlukování pomocí takzvaných siluet. Teoretické části těchto kapitol doplňují ilustrační příklady.

V kapitole 4 je představen program pro shlukovou analýzu, který v rámci této práce vznikl. Jsou popsány jeho základní funkce a uživatelské rozhraní.

Vzniklý software je následně použit pro shlukování reálných dat z nástroje NERUDA. Obdržené výsledky jsou prezentovány a vyhodnoceny v kapitole 5. Nejprve je shlukování provedeno pouze pro zařízení Zlínského kraje. Nejlepších výsledků je dosaženo algoritmem k-means, který z celkového množství 5000 scénářů vytvoří 34 shluků, z nichž každý lze reprezentovat jedním scénářem. Tím dojde k významné redukci scénářů pro analýzu nástrojem PIGEON a úspoře času.

Následuje shlukování z hlediska proměnných celé ČR. Obdržené výsledky však mají kvůli prokletí dimenzionality minimální vypovídající hodnotu.

Do budoucna by tedy bylo vhodné prozkoumat metriky, které tomuto jevu nepodléhají. S tím souvisí i nutnost použití algoritmů, které s těmito vzdálenostmi dokáží pracovat. Jako nejvhodnější se v tomto případě jeví metoda k-medoids. Jedná se o obdobu k-means, shluky však nejsou reprezentovány jejich těžištěm, ale jedním z bodů shluku. Výhodou tohoto přístupu je i to, že tento reprezentant je metodou automaticky určen, a nemusíme ho tak vybírat pomocí siluet.

SEZNAM POUŽITÉ LITERATURY

[1] AGGARWAL, Charu C., Alexander HINNEBURG a Daniel A. KEIM. On the Surprising Behavior of Distance Metrics in High Dimensional Space. *Lecture Notes in Computer Science*. , 420-434. DOI: 10.1007/3-540-44503-X_27. Dostupné také z: http://link.springer.com/10.1007/3-540-44503-X_27

[2] ANDERBERG, Michael R. *Cluster analysis for applications*. New York: Academic Press, 1973, xiii, 359 p. ISBN 01-205-7650-3.

[3] BEZDEK, James C. *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Springer Verlag, 2013, xv, 256 p. ISBN 978-147-5704-525.

[4] EVERITT, Brian. *Cluster Analysis*. 5th ed. Chichester: Wiley, 2011, xii, 330 s. Wiley series in probability and mathematical statistics, 848. ISBN 978-0-470-74991-3.

[5] GAN, Guojun. *Data clustering: theory, algorithms, and applications*. Philadelphia: SIAM, Society for Industrial and Applied Mathematics, c2007, xxii, 466 s. ISBN 978-0-898716-23-8.

[6] JAIN, Anil K a Richard C DUBES. *Algorithms for clustering data*. Englewood Cliffs, N.J.: Prentice Hall, c1988, xiv, 320 p. ISBN 01-302-2278-X.

[7] LANCE, G. N. a W. T. WILLIAMS. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*. 1967, **9**(4), 373-380. DOI: 10.1093/comjnl/9.4.373. ISSN 0010-4620. Dostupné také z: http://comjnl.oxfordjournals.org/cgi/doi/10.1093/comjnl/9.4.373

[8] LUKASOVÁ, Alena a Jana ŠARMANOVÁ. *Metody shlukové analýzy*. Praha: Státní nakladatelství technické literatury, 1985, 210 s.

[9] ROUSSEEUW, Peter J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*. 1987, **20**, 53-65. DOI: 10.1016/0377-0427(87)90125-7. ISSN 03770427. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/0377042787901257

[10] SCHWAMMLE, V. a O. N. JENSEN. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. *Bioinformatics*. 2010, **26**(22), 2841-2848. DOI: 10.1093/bioinformatics/btq534. ISSN 1367-4803. Dostupné také z: http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btq534

[11] ŠOMPLÁK, Radovan et al. Multi-Commodity Network Flow Model Applied to Waste Processing Cost Analysis for Producers. In: *Chemical Engineering Transactions*. 45. 2015, s. 733-738. DOI: 10.3303/CET1545123.

[12] ŠOMPLÁK, Radovan. *Efektivní plánování investic do technologií pro energetické využití odpadů*. Brno, 2016, 108 s. Dizertační práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Prof. Ing. Petr Stehlík, CSc., dr. h. c.

SEZNAM PŘÍLOH

A SILUETY PRO 34 SHLUKŮ Z HLEDISKA ZLÍNSKÉHO KRAJE